

はじめてのアプリ作成 ブブゼラアプリを作ってみた

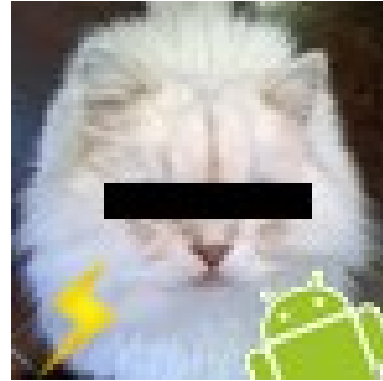
株式会社ブリリアントサービス
要 徳幸

自己紹介

- 名前:
 要 徳幸 (カナメ ノリュキ)
- 所属:
 株式会社ブリリアントサービス 技術部
- 開発経歴
 携帯開発組み込み(Brew、Symbian)
- TwitterID: nory_kaname



今日は・・・

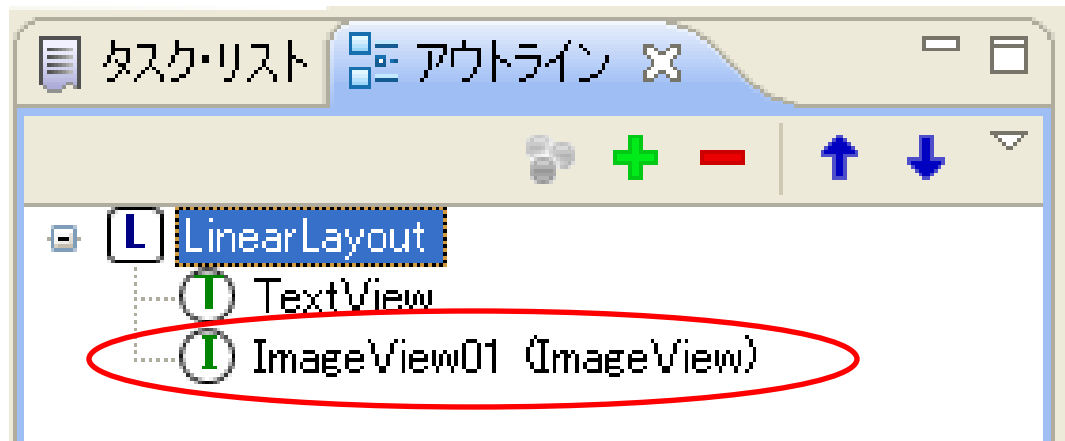


アプリを作ってみる

- なにかアプリを作ってみる
- 目標(やりたいこと)
 - イメージ画像を貼り付ける
 - 音を鳴らしてみる
 - 音量を扱ってみる
 - なにか標準widgetを使ってみる

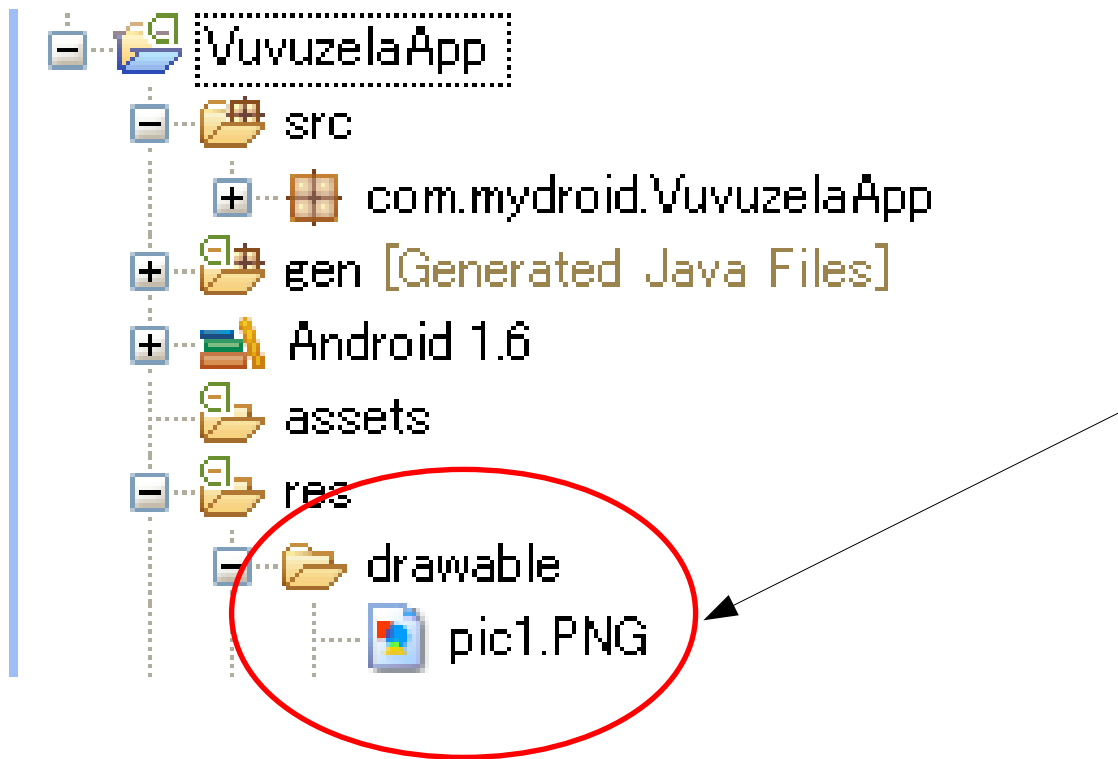
ブブゼラの画像を表示

- レイアウトファイルのMain.xmlに「ImageView」を追加する。



ブブゼラの画像を表示

- 画像のリソースを追加
 - Res/drawableの下に追加する



ブブゼラの画像を表示

- 実装方法
 - ImageViewのwidget取得
 - リソースファイルをセット
- プログラムは

```
/* 変数宣言 */  
private ImageView myImageView;  
private int myImageId;  
/* プログラム */  
myImageView =  
(ImageView)findViewById(R.id.ImageView01);  
myImageId = R.drawable.wc1;  
myImageView.setImageResource(myImageId);
```

ブブゼラの画像を表示

- 実行結果、みごとに画像が表示されました！

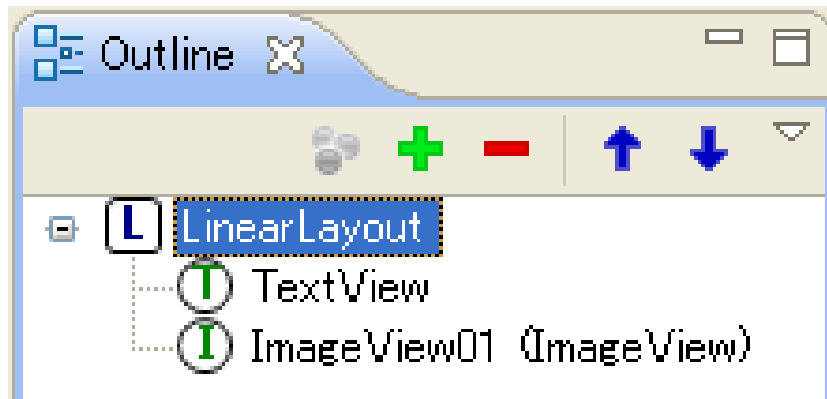


ブブゼラの画像を表示

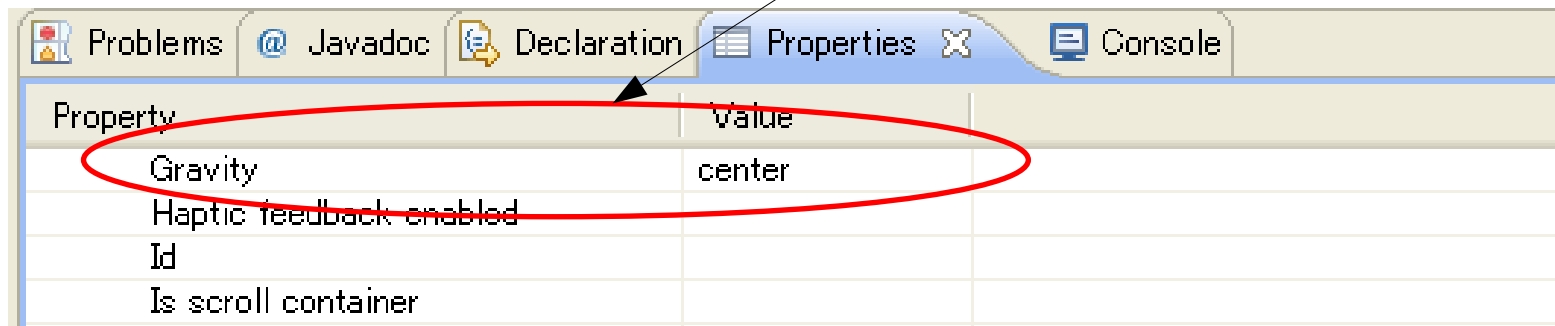
- でも、画像位置が・・・
 - 画面中央に表示されるように調整しよう～



ブブゼラの画像を表示



プロパティでGravityを「center」に指定する。

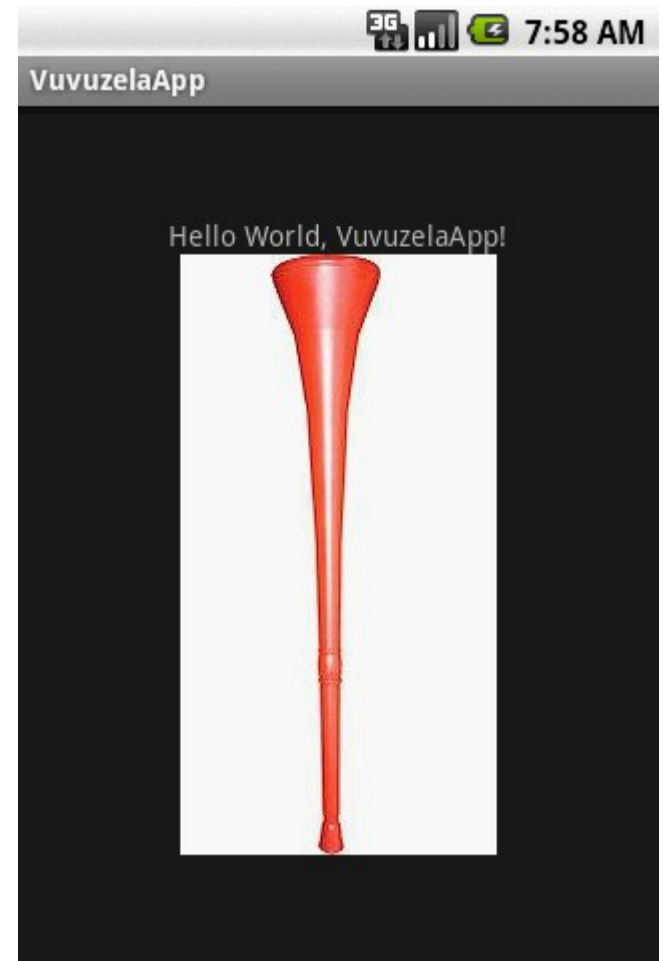


ブブゼラの画像を表示

- 指定なし



- Center指定

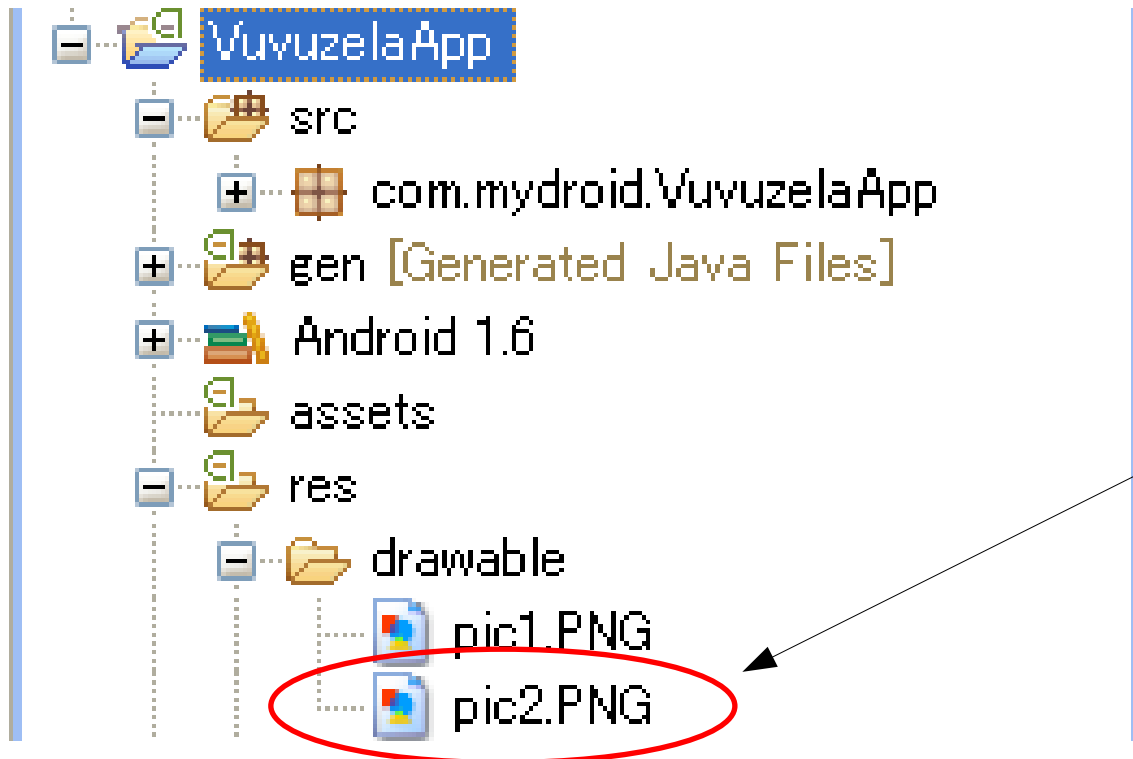


次の試み・・・

- でも、これだけじゃおもしろくないなあ
 - 画像を触ったら、ブブゼラが鳴っている画像にすり替えよ～

画像のすり替え

- 画像のリソースを追加
 - res/drawableの下に「pic2.PNG」を追加

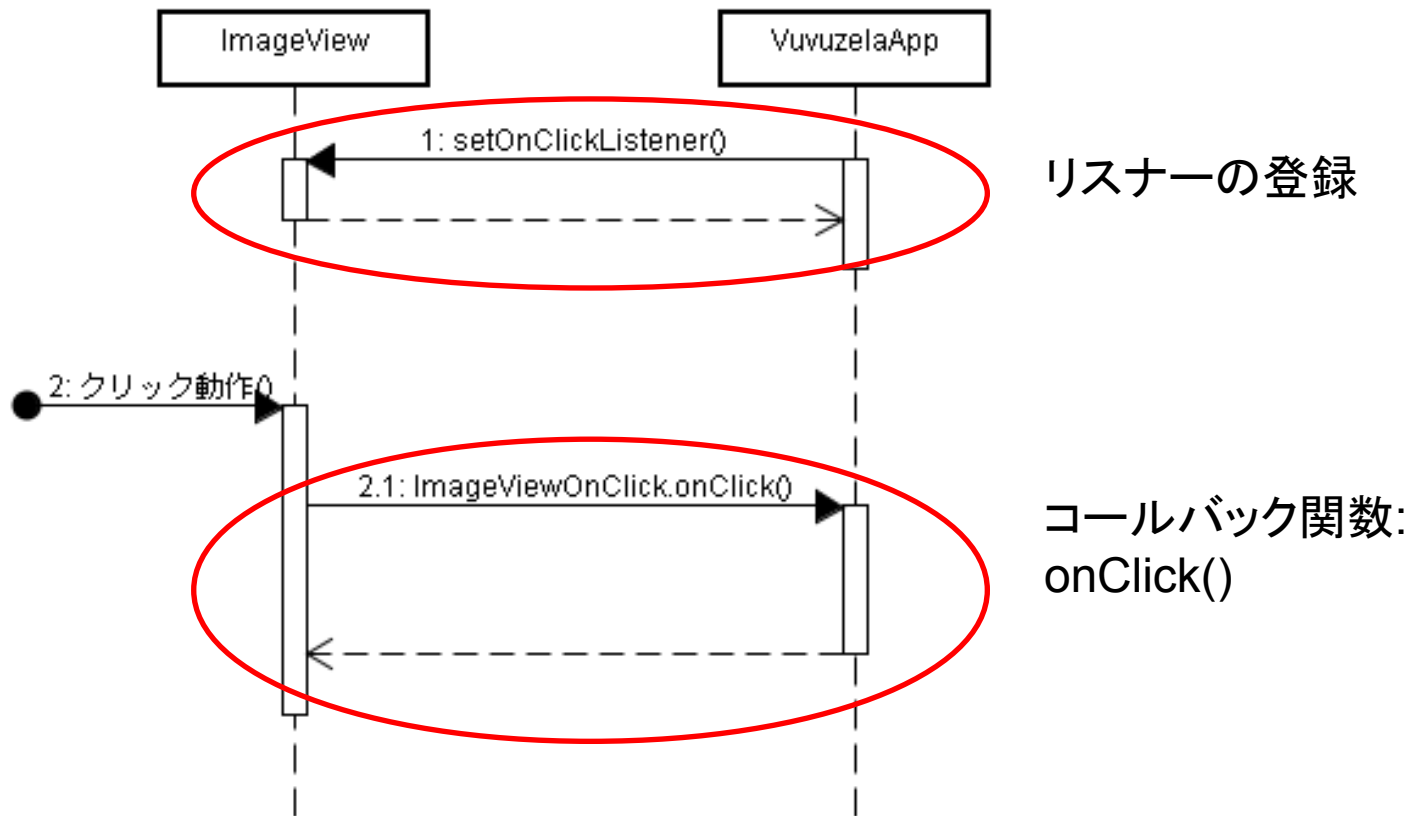


画像のすり替え

- どうやったらクリックイベントが拾えるか
- 実装方法
 - ImageView用のonClickListener()を実装する
 - ImageViewにonClickListener()を登録する

画像のすり替え

- シーケンス図



画像のすり替え

- プログラム

```
/*ClickListenerの登録*/
myImageView.setOnClickListener(new ImageViewOnClick() );

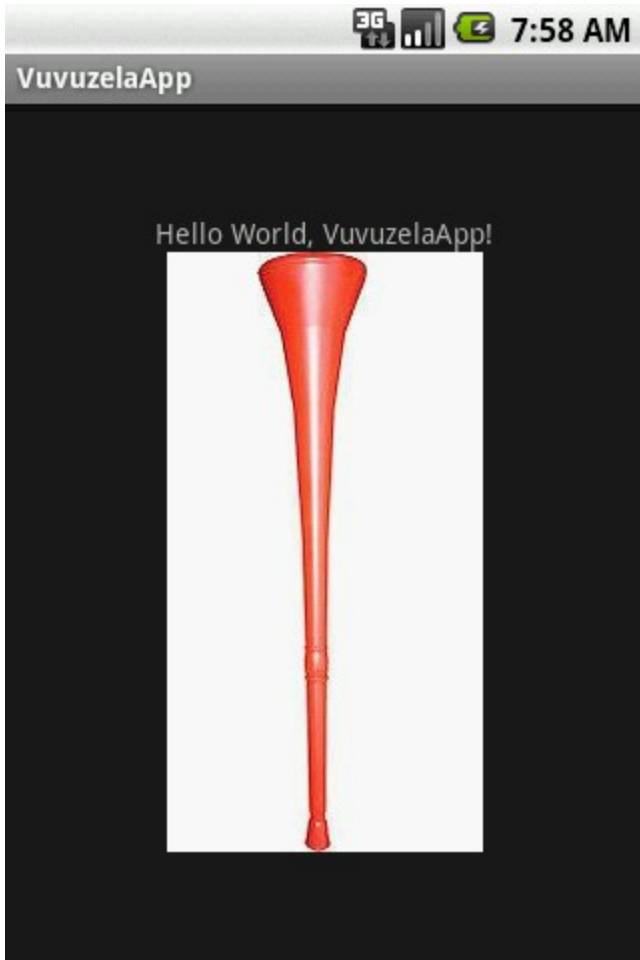
/* ClickListenerの実装 */
public class ImageViewOnClick implements OnClickListener {

    @Override
    public void onClick(View arg0) {
        // TODO 自動生成されたメソッド・スタブ

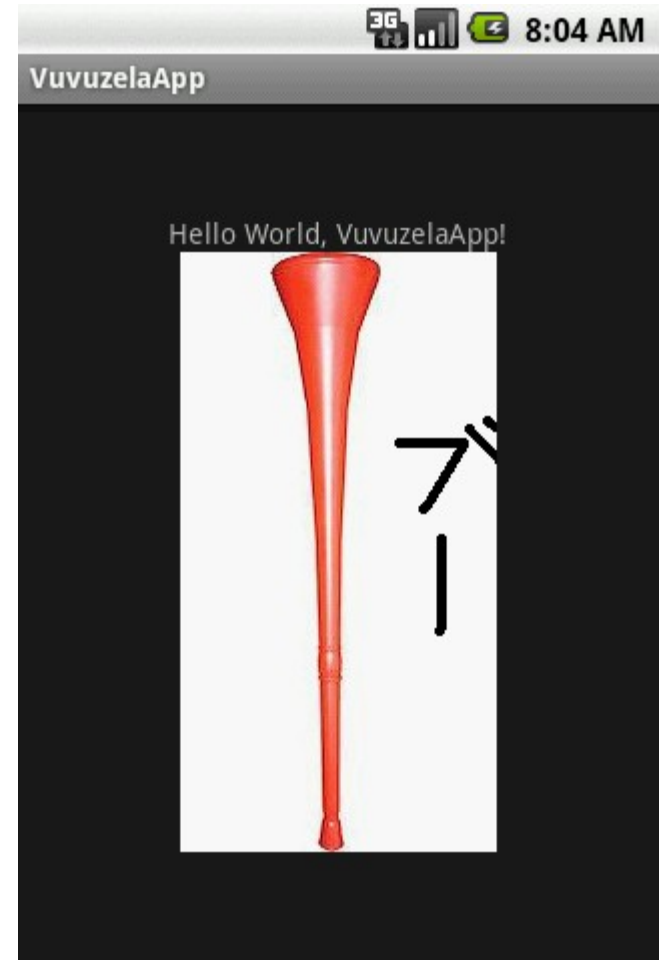
        /* ここに画像切り替え処理を実装 */

    }
}
```


画像のすり替え




画像を触ると



次の試み

- まだ、これだけじゃおもしろくないなあ
– ブブゼラの「ブーーーーー」という音でもつけてみよ

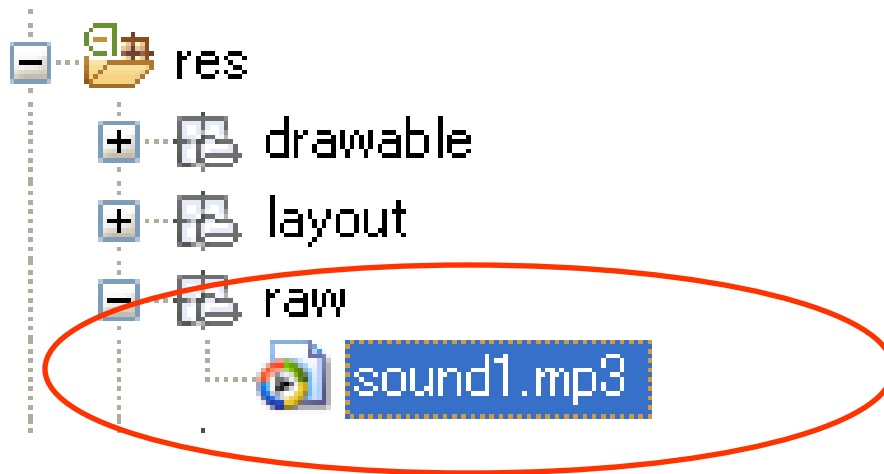
 ブオオオオオオオオオオ！))

音を鳴らしてみる

- 音の再生方法
- 実装方法(必要なこと)
 - 音源ファイルを用意する
 - 音源の再生処理を追加
 - 音源の停止処理を追加
 - 音源ファイル再生終了後、画像を1枚目に戻す

音の再生の準備

- 音源ファイルを用意する
 - resディレクトリに、rawディレクトリを作成し、res/rawディレクトリに 保存する



音源の再生

- MediaPlayerクラスを使用する
 - createでインスタンス生成
 - seekToで再生位置指定
 - startで再生開始

```
/* 変数宣言 */  
private MediaPlayer mMP;  
private int mySound;  
/* プログラム */  
mySound = R.raw.sound1;  
mMP = MediaPlayer.create(this, mySound);  
mMP.seekTo(0);  
mMP.start();
```

音源の停止

- MediaPlayerクラスを使用する
 - Stopで再生停止

```
/* 停止 */  
mMP.stop();
```

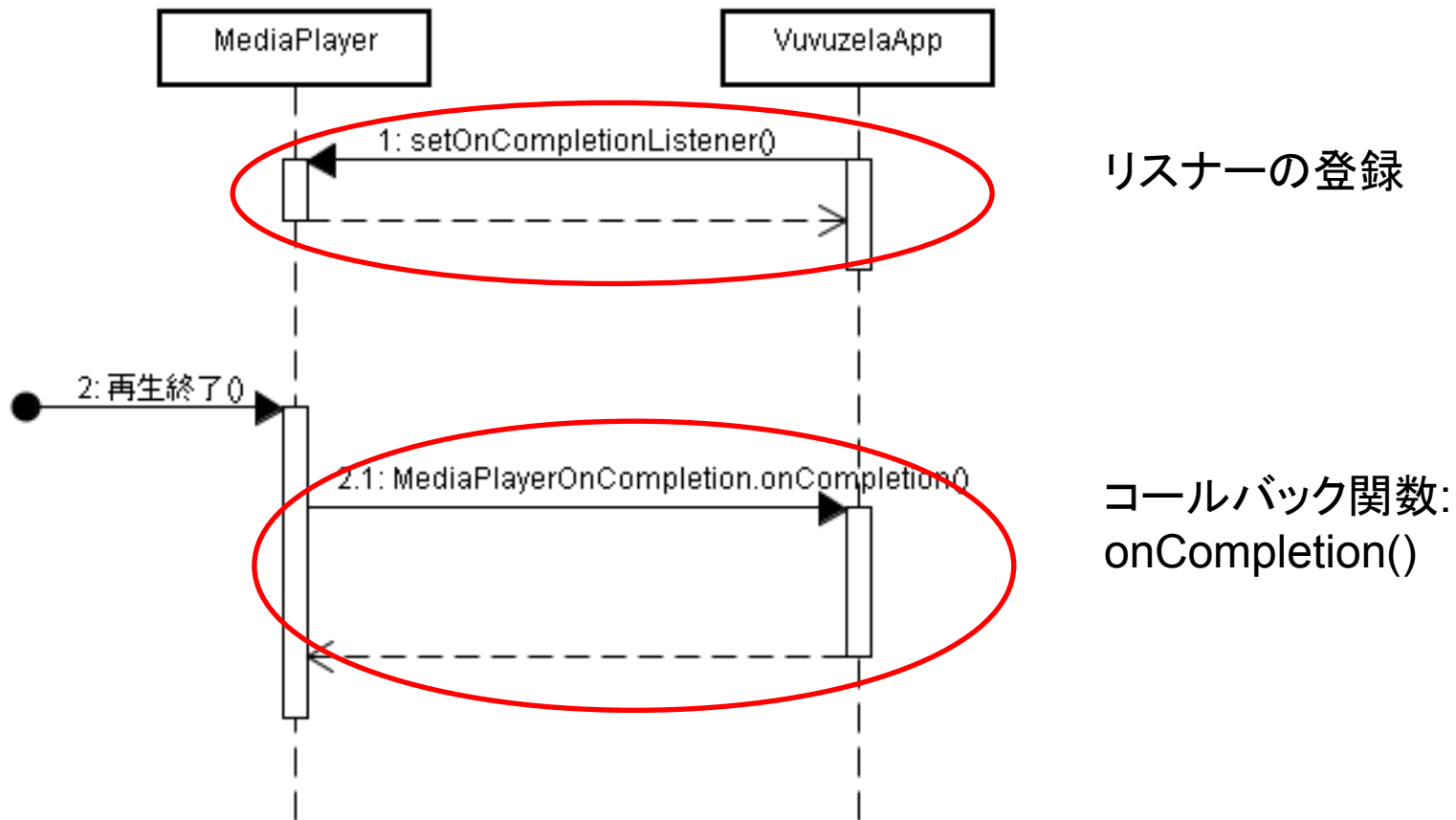
音源ファイルの再生終了

- MediaPlayerクラスを使用する
 - setOnCompleteListenerにコールバック関数をセットすることで、再生終了後に呼ばれる

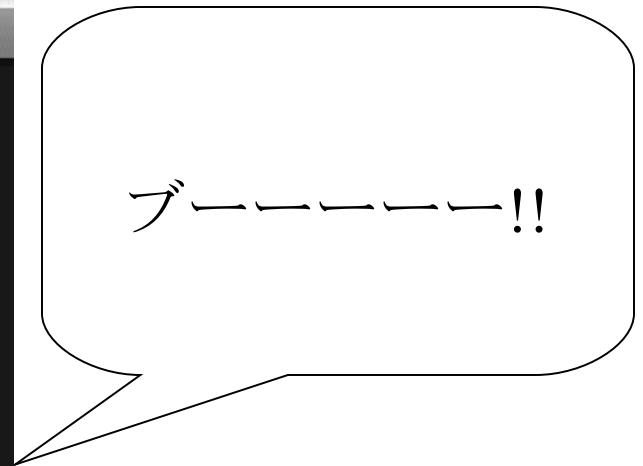
```
/* コールバック関数を登録 */  
mMP.setOnCompleteListener(new MediaPlayerOnCompletion() );  
  
/* コールバック関数を実装 */  
public class MediaPlayerOnCompletion implements OnCompleteListener  
{  
    @Override  
    public void onCompletion(MediaPlayer arg0) {  
        myImageId = R.drawable.pic1;  
        myImageView.setImageResource(myImageId);  
    }  
}
```

音源ファイルの再生終了

- シーケンス図

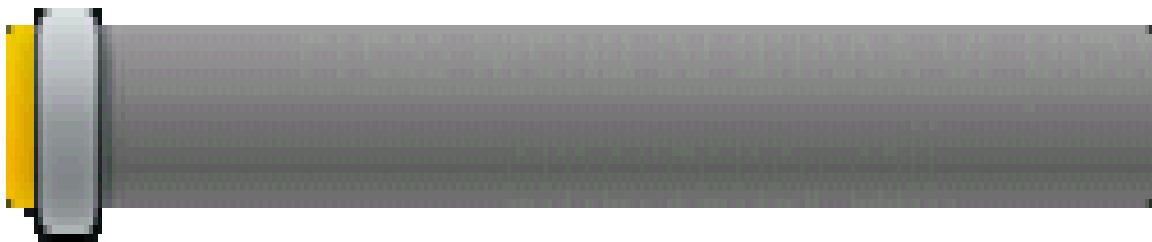


実行結果



次の試み

- 音量設定を行えるようにSeekBarを設置してみよう
 - 動的に「ブー」の音量を設定できるようにする



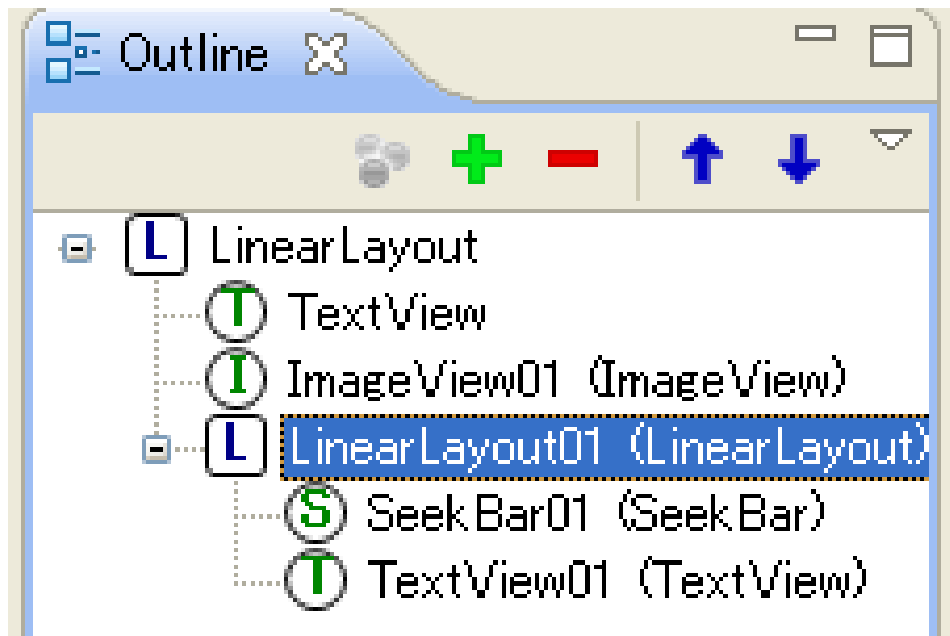
SeekBarの設置

- こんな感じで設置したい
 - LinearLayoutの中にSeekBarとTextViewを配置



SeekBarの設置

- 実際にLayoutファイル、main.xmlを編集



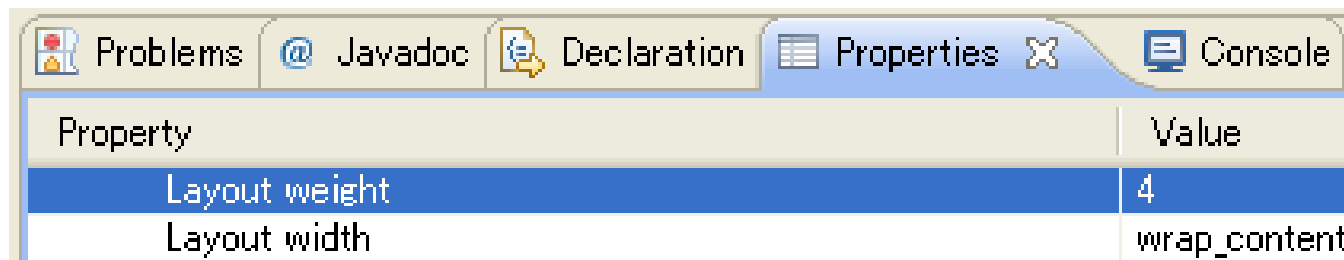
実行してみると・・・



SeekBarがなんか小さい・・・
バランスが悪いなあ。

SeekBarの設置(整える)

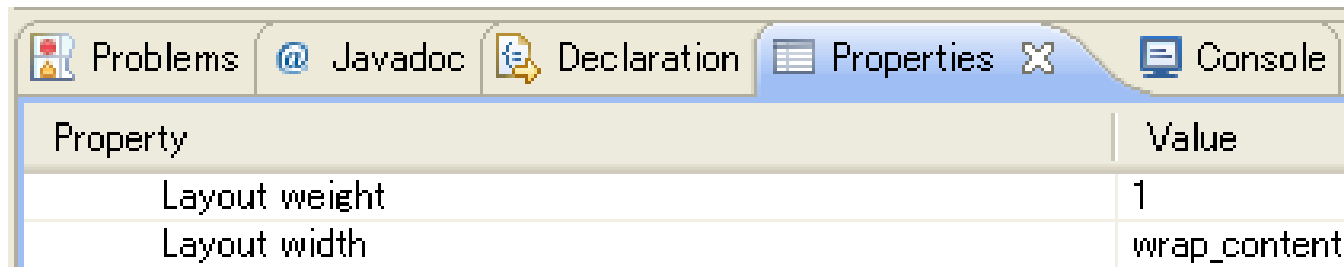
- プロパティのweightパラメータを調整する
 - SeekBarの設定



The screenshot shows the Properties view in an IDE. The 'Properties' tab is active, and the table below lists the properties of the selected widget. The 'Layout weight' property is highlighted in blue and has a value of 4.

Property	Value
Layout weight	4
Layout width	wrap_content

- textViewの設定



The screenshot shows the Properties view in an IDE. The 'Properties' tab is active, and the table below lists the properties of the selected widget. The 'Layout weight' property has a value of 1.

Property	Value
Layout weight	1
Layout width	wrap_content

実行してみると・・・



いい感じで表示されました！

ついでに、TextViewには

- Gravityに「center」

- Textに「1」

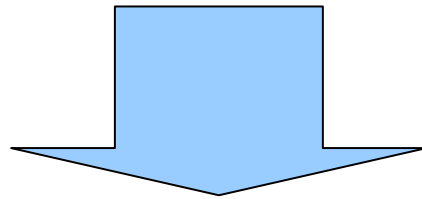
を設定

SeekBarの情報セット

- 音量のMAX値を取得してSeekBarへセット
- 現在の音量を取得してSeekBarへセット
- SeekBarを操作した際のコールバック関数
セット

AudioManagerから音量取得

- 音の音量を管理しているクラス、AudioManagerから音量を取得する
 - MAX値 `getStreamMaxVolume()`
 - 音量 `getSteamVolume()`

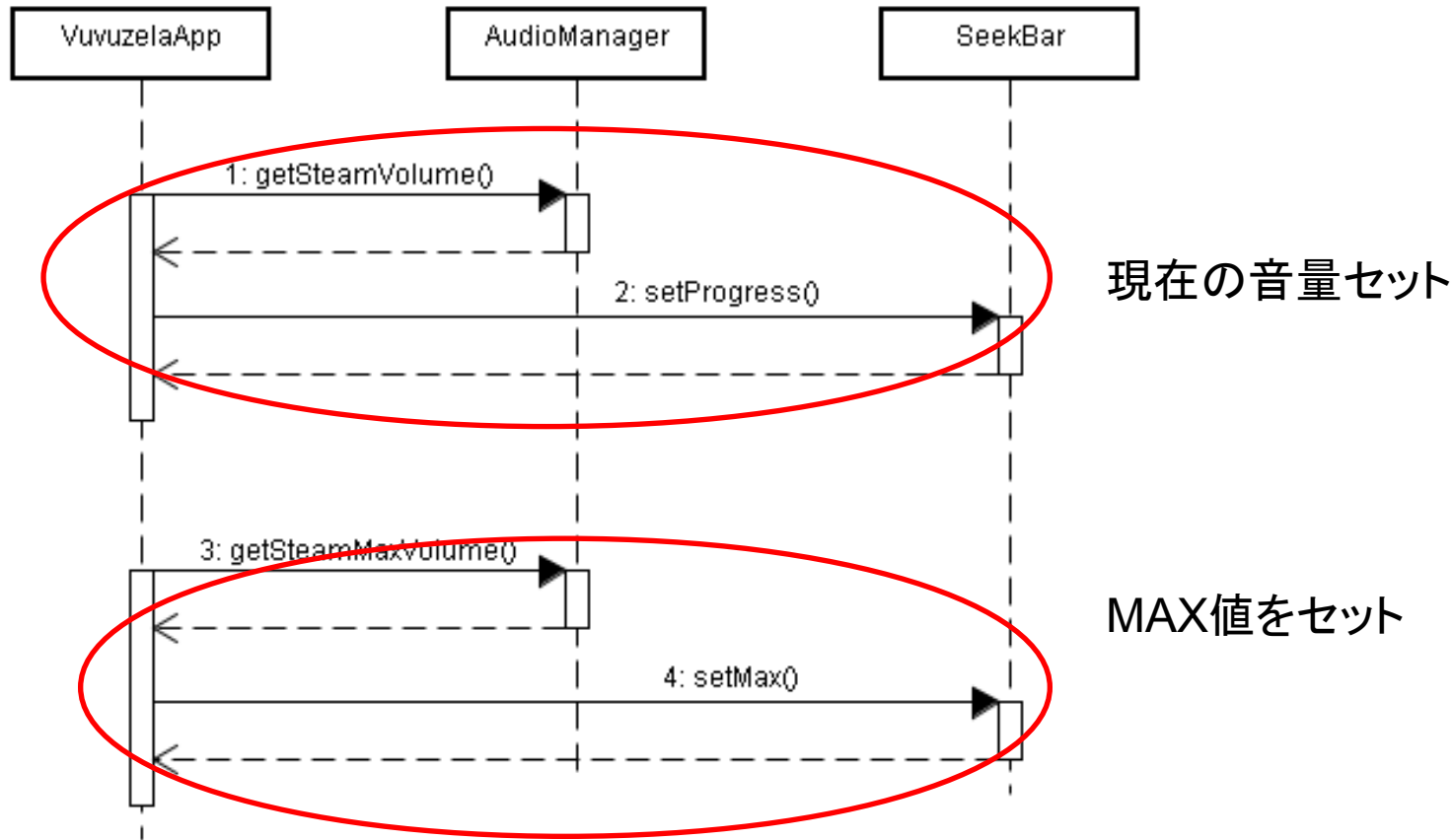


この値をSeekBarのパラメータに
セットする

SeekBarのパラメータへセット

- MAX値をセット
 - SeekBar.setMax()にてセット
- 現在の音量値をセット
 - SeekBar.setProgress()にてセット

SeekBarのパラメータへセット



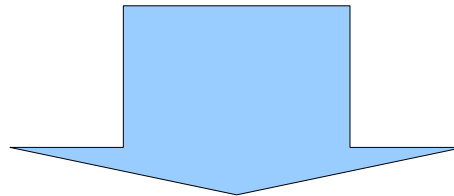
実行してみると



MAX = 15 と
現在値 = 13 がセットさ
れました

SeekBarを操作

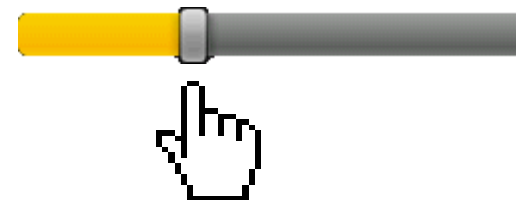
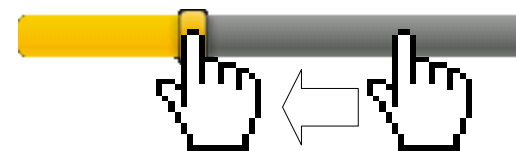
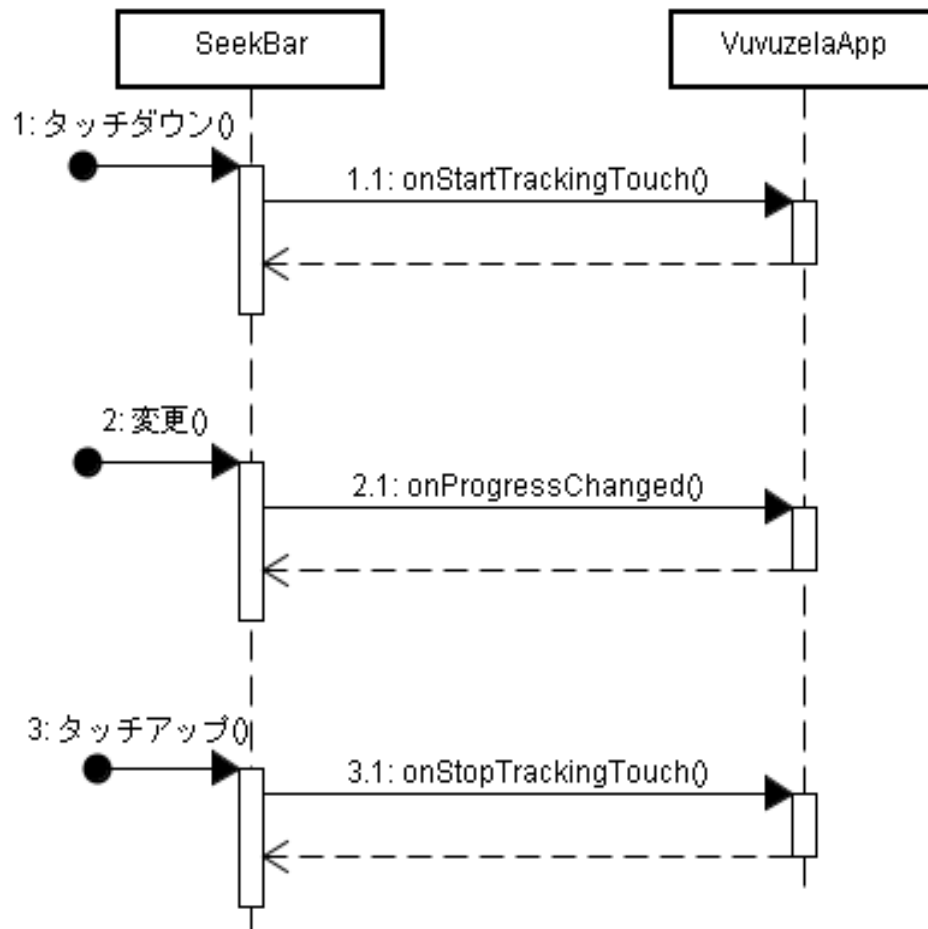
- SeekBarを操作した際のイベントリスナーを登録
 - アプリのクラス自身にimplementsで追加する



```
public class VuvuzelaApp extends Activity
    implements OnSeekBarChangeListener {
```

SeekBarを操作

- シーケンス図



SeekBarを操作



まとめ

目標(やりたかったこと)

- イメージ画像を貼り付ける
 - ImageViewを使用
- 音を鳴らしてみる
 - MediaPlayerを使用
- 音量を扱ってみる
 - AudioManagerを使用
- なにか標準widgetを使ってみる
 - SeekBarを使用

ご清聴ありがとうございました

あと一つだけ・・・

Android Hacks発売

Android Hacks

プロが教えるテクニック & ツール

2010年09月03日 発売予定

定価3,990円

株式会社ブリリアントサービス 著

